VirtualButcher: Coarse-to-fine Annotation Transfer of Cutting Lines on Noisy Point Cloud Reconstruction

Raphael Falque[1,†], Teresa Vidal-Calleja[1], Malcolm McPhee[2], Edwina Toohey[2], and Alen Alempijevic[1]

***Abstract*—Robotics and automation are rapidly becoming part of meat processing operations. Current automation of breaking down a carcass into primals relies on guidance from X-ray, interconnected with robotised band-saws. While yielding very accurate cutting lines, the use of vision systems for guidance would be significantly more affordable. This work proposes a novel method that solves the annotation transfer between a 3D noise-free cutting line annotated on a CT acquired canonical model and a noisy target in the form of a point cloud acquired by RGB-D cameras. The proposed coarse-to-fine method initially aligns the posture of each body using a non-rigid deformation algorithm and then performs a local search to solve the surface correspondence which is later used to morph the template non-rigidly. We quantitatively assess the approach by benchmarking with multiple state-of-the-art algorithms on a public available human pose dataset. We also present a proof of concept evaluation on lamb carcasses.**

Keywords: annotation transfer, skeletonization, non-rigid deformation, cutting lines, RGB-D cameras.

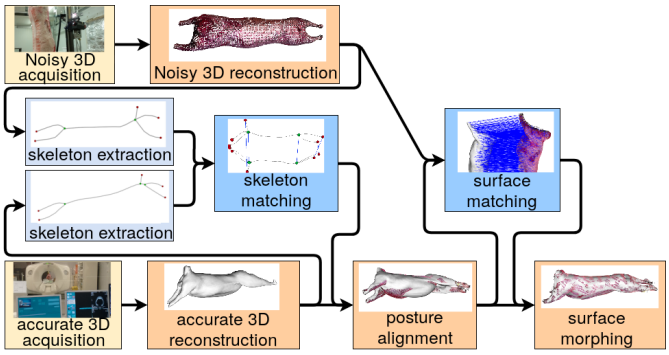


Fig. 1: Overview of the proposed approach: the template and target skeletonization are used for a coarse alignment. The annotation transfer is then solved by morphing the template into the target.

I. Introduction

Abattoirs are moving the processing of lamb carcasses towards automation, which allows for faster, precise, and efficient processing [1]. While improving the carcass value, this also reduced human contact with carcasses. In particular, in the context of the cutting of the carcasses, the automation allows eliminating potentials hazards to meat process workers [2].

The current automation of cutting carcass primals relies on technology such as X-ray or DEXA [1] to guide the blades which perform the cuts. While providing a very accurate guideline for the process automation, these systems are expensive to set up and require heavy infrastructures to protect the workers from radiations.

This paper argues that an affordable and non-invasive system using off-the-shelf 3D sensors can be built as a good alternative to such systems. Our proposed algorithm consider the 3D reconstruction from RGB-D cameras [3] of a lamb carcass as an input and can predict the cutting lines that should be followed by a rotary band-saw. The integration of such a system in an industrial process would then require minimal modifications to a production line.

The automation of finding where to cut a carcass can be obtained by solving the annotation transfer between an annotated template and the 3D reconstruction of a carcass that needs to be cut. This paper proposes a robust coarse-to-fine systematic approach for morphing a template into a target (and subsequently solving the annotation transfer). We show that, if possible, decomposing the morphing as a two-step process by firstly aligning the posture of the template with the target and then tackling the surface alignment as a local search gives better performance than more general state-of-the-art methods of surface correspondence. The algorithmic contributions of this paper are an improved deformation method that builds upon embedded deformation (ED) [4] and a correspondences filtering method inspired by as-rigid-as-possible (ARAP) deformation [5]. An overview of the proposed approach is shown in Fig. 1.

† Raphael.Guenot-Falque@uts.edu.au. [1] Robotics Institute (UTS:RI) at the University of Technology Sydney (UTS), Australia. [2] NSW Department of Primary Industries, Australia.

II. Method

Let us assume a noise-free mesh as a template $\mathcal{S}_1$ and a noisy point cloud or a mesh, acquired by RGB-D cameras and a 3D reconstruction algorithm [3], as a target $\mathcal{S}_2$. To simplify the complexity of the problem, we decompose our approach in two parts: A) a robust prior is first obtained by aligning the posture of both shapes, and B) surfaces are then aligned by performing a local search between both surfaces.

A. Coarse posture alignment

The aim of this first part is to find a good prior for the surface matching by aligning the posture of both shapes. In this section, a skeleton-like is extracted from both shapes $\mathcal{S}_1$ and $\mathcal{S}_2$. Several features nodes from the skeletons are then selected and associated. These associations are then finally used to align both shapes using a non-rigid deformation.

1) Skeleton extraction: The skeleton of a 3D surface is a 1D curvilinear structure that describes its essential topology. The skeletonisation method used here follows [6] for the template (mesh skeletonisation), and [7] for the targets (pointcloud

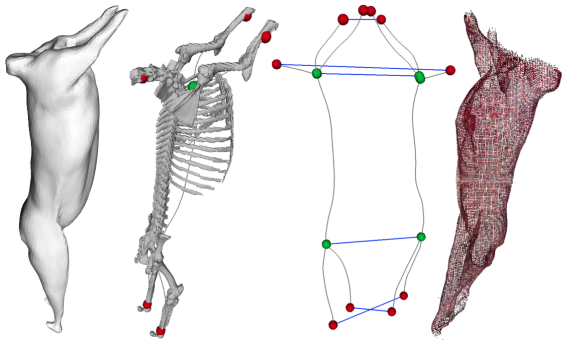


Fig. 2: Skeleton extraction and matching between the template and a target (with the template on the left, and the noisy target on the right). The two skeletons are matched using Eq. (3).

skeletonisation). The mean curvature flow (MCF) skeletonization gets as input the mesh $\mathcal{S}_i$ and produces a skeleton $\mathcal{Q}_i$ which is defined as a set of nodes and a set of edges that connect nodes together. As shown in Fig. 2, the MCF skeleton has a close correspondence with the biological skeleton (which was obtained using computerized axial tomography (CT)).

2) Skeleton feature nodes association: After the skeletonization step, we perform a pairwise association between feature nodes. From all the vertices of the skeleton, the set of *feature nodes* is defined by both *terminal nodes* and *junction nodes*. The terminal nodes correspond to the extremity of the skeleton (in red in Fig. 2) and the junction nodes correspond to the nodes linking several branches of the skeleton (in green in Fig. 2). The set of junction nodes $\mathcal{J}_i$ is defined as:

$$\mathcal{J}_i = \{\boldsymbol{a}_{i,1}, ..., \boldsymbol{a}_{i,n}\} \mid \boldsymbol{a}_{i,j} \in \mathcal{Q}_i \text{ and } \kappa(\boldsymbol{a}_{i,j}) \geq 3, \quad (1)$$

and the set of terminal nodes $\mathcal{T}_i$ as:

$$\mathcal{T}_i = \{\boldsymbol{a}_{i,1}, ..., \boldsymbol{a}_{i,m}\} \mid \boldsymbol{a}_{i,j} \in \mathcal{Q}_i \text{ and } \kappa(\boldsymbol{a}_{i,j}) = 1, \quad (2)$$

where $\kappa(.)$ is the connectivity, n is the cardinality of $\mathcal{J}_i$, m is the cardinality of $\mathcal{T}_i$, $\boldsymbol{a}_{i,j} \in \mathbb{R}^3$ are vertices of the skeleton $\mathcal{Q}_i$ (in the following part of the paper we refer to feature nodes $\mathcal{A}_i$ that are either junction nodes or terminal nodes). Thereafter, skeletons are matched by defining a set of pairwise associations for every node. We build the exhaustive set of possible feature correspondences between $\mathcal{T}_1$ and $\mathcal{T}_2$ alongside with the combinations of $\mathcal{J}_1$ and $\mathcal{J}_2$ and trim all the impossible combinations where a node would be associated twice. Afterward, we exhaustively search for the set of correspondences that minimizes

$$\underset{\boldsymbol{R},\boldsymbol{p}}{\operatorname{argmin}} \sum_{l=1}^{n+m} \left\| (\boldsymbol{R}\, \boldsymbol{a}_{1,l} + \boldsymbol{p}) - \boldsymbol{a}_{2,l} \right\|^2, \quad (3)$$

with $\boldsymbol{R} \in \mathbb{SO}(3)$ and $\boldsymbol{p} \in \mathbb{R}^3$ respectively the rotation and the translation that minimize the distance between the set of feature nodes $\{(\boldsymbol{a}_{1,1}, \boldsymbol{a}_{2,1}), ..., (\boldsymbol{a}_{1,n+m}, \boldsymbol{a}_{2,n+m})\}$. The set of pairwise associations with the smaller residuals in Eq. (3) is then selected as the final pairwise correspondences of the skeleton features. An example of the feature node association is illustrated in Fig. 2.

Algorithm 1: Return $\mathcal{N}(\boldsymbol{g})$, the μ nearest neighbors of $\boldsymbol{g}$ in $\mathcal{G}$ by exploring the mesh $(\mathcal{V}, \mathcal{F})$.

Input: $\mathcal{V}$, $\mathcal{F}$, $\mathcal{G}$, $\boldsymbol{g}$, μ
Output: $\mathcal{N}(\boldsymbol{g})$ s.t., $|\mathcal{N}(\boldsymbol{g})| = \mu$

function Search Neighbors
  for $i = 1$ *to* $|\mathcal{V}|$ **do**
    dist$[i] \leftarrow$ inf
  $\mathcal{V}_{\text{visited}} \leftarrow \emptyset$, $\mathcal{V}_{\text{to\_visit}} \leftarrow \emptyset$, $\mathcal{N}(\boldsymbol{g}_i) \leftarrow \emptyset$
  current $\leftarrow$ index of $\boldsymbol{g}$ in $\mathcal{V}$
  dist[current] $\leftarrow 0$
  nodes_found $\leftarrow 0$
  while *nodes_found* $\neq \mu$ **do**
    foreach $\boldsymbol{v}_j \in \mathcal{N}(\boldsymbol{v}_{current})$ **do**
      if $\boldsymbol{v}_j \notin (\mathcal{V}_{visited} \cup \mathcal{V}_{to\_visit})$ **then**
        $\mathcal{V}_{\text{to\_visit}} \leftarrow \mathcal{V}_{\text{to\_visit}} \cup \boldsymbol{v}_j$
      $d \leftarrow \|\boldsymbol{v}_j - \boldsymbol{v}_{\text{current}}\|_2$
      if *dist*$[j] >$ *dist[current]*$+d$ **then**
        dist$[j] \leftarrow$ dist[current]$+d$
    $\mathcal{V}_{\text{visited}} \leftarrow \mathcal{V}_{\text{visited}} \cup \boldsymbol{v}_{\text{current}}$
    if $\boldsymbol{v}_{current} \in \mathcal{G}$ **then**
      nodes_found $\leftarrow$ nodes_found $+1$
      $\mathcal{N}(\boldsymbol{g}) \leftarrow \mathcal{N}(\boldsymbol{g}) \cup \boldsymbol{v}_{\text{current}}$
    current $\leftarrow$ argmin(dist$(\mathcal{V}_{\text{to_visit}})$)
    $\mathcal{V}_{\text{to\_visit}} \leftarrow \mathcal{V}_{\text{to\_visit}} \setminus \boldsymbol{v}_{\text{current}}$
  return $\mathcal{N}(\boldsymbol{g})$

3) Non-rigid deformation: Once the skeleton feature nodes have been associated, the template $\mathcal{S}_1$ is deformed to match the posture of the target $\mathcal{S}_2$ using a modified version of ED [4] algorithm. In brief, ED performs the non-rigid deformation by optimizing a deformation graph $(\mathcal{G}, \mathcal{E})$ which is defined as an undirected graph with nodes set $\mathcal{G} = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_\nu\}$ (the nodes are sampled from $\boldsymbol{v}_1$) and edge set $\mathcal{E} \subseteq \mathcal{G} \times \mathcal{G}$. The mesh vertices are then updated w.r.t. the nodes of the deformation graph in their neighborhood.

Our approach differs from the method proposed by Sumner *et al.* [4] in the building process of the deformation graph. In [4], the set $\mathcal{E}$ is built by searching the μ closest point of each node using a K-NN search. However, the use of the Euclidean distance can lead to ill-connected deformation graph (*e.g.*, , the legs of an animal could be constrained in the deformation if the feet are too close). In our case, as the template is defined as a mesh, we can build the edges of the graph by using the geodesic distance. To do so, we search the neighbors of each node in a Dijkstra-style fashion by using the original mesh as the space to explore[1].

The formal definition of our proposed search method is given in Algorithm 1. This differs from the Dijkstra algorithm [8] in the sense that we explore the graph until μ nodes

[1]A consequence of the search being performed on $\mathcal{S}$, is that $\mathcal{G}$ has to be a subset of $\mathcal{V}$ and not a simple topological approximation of the shape (which would be the case for voxel grid downsampling).

are found, and we do not keep track of the path as it is not relevant here. One might argue that the exact geodesic distance [9] or an approximation of this distance such as heat kernel distances [10] should be used instead. However, after empirical evaluation of both methods, we concluded that the former is too costly from a computational point of view, and the latter can yield outliers.

Once the modified deformation graph is created, we search the set of local rotations $\boldsymbol{R}_j^d \in \mathbb{R}^{3\times3}$ and local translations $\boldsymbol{p}_j^d \in \mathbb{R}^3$ for every node $\boldsymbol{g}_j$ of $\mathcal{G}$. An energy function that accounts for the pairwise distance between features nodes E_{con}, the rotation E_{rot}, and the regularization E_{reg} is defined as:

$$\underset{\boldsymbol{R}_1^d,\boldsymbol{p}_1^d,\ldots,\boldsymbol{R}_\nu^d,\boldsymbol{p}_\nu^d}{\operatorname{argmin}} \quad w_{con}E_{con} + w_{rot}E_{rot} + w_{reg}E_{reg} + w_{rig}E_{rig} \tag{4}$$

with E_{con} defined with the skeleton feature nodes such as

$$E_{con} = \sum_{l=1}^{n+m} \|\boldsymbol{a}_{1,l} - \boldsymbol{a}_{2,l}\|_2^2 . \tag{5}$$

E_{rot} adds the errors of all the rotations matrices and is defined similarly to [11] as

$$E_{rot} = \sum_{j=1}^{\nu} \left\| \boldsymbol{R}_j^{dT}\boldsymbol{R}_j^d - I \right\|_F^2 . \tag{6}$$

The regularization term prevents divergence of the neighboring nodes exerts on the overlapping space,

$$E_{reg} = \sum_{i=1}^{\nu}\sum_{j=1}^{\mu} \left\| \boldsymbol{R}_j^d(\boldsymbol{g}_i - \boldsymbol{g}_j) + \boldsymbol{g}_j + \boldsymbol{p}_j^d - (\boldsymbol{g}_i + \boldsymbol{p}_i^d) \right\|_2^2 . \tag{7}$$

Similarly to Chen *et al*. [11], we add an additional regularization term E_{rig} on the *bridges* in the deformation graph defined as

$$E_{rig} = \sum_{(j,k)\in\mathcal{E}_b} \left\| \boldsymbol{R}_j^{dT}\boldsymbol{R}_k^d - I \right\|_F^2 , \tag{8}$$

with $\mathcal{E}_b$, the set of bridges present in the deformation graph (found using depth-first search (DFS) [12]).

The energy function described in (4) is then minimized with a Levenberg-Marquardt optimization. Once the new position of the deformation graph nodes is known, the vertices of $\mathcal{S}_1$ are updated using

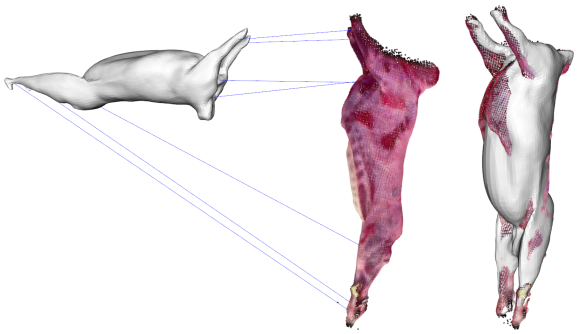
$$\boldsymbol{v}_i^* = \sum_{j=1}^{\mu} w_j(\boldsymbol{v}_i)[\boldsymbol{R}_j^d(\boldsymbol{v}_i - \boldsymbol{g}_j) + \boldsymbol{g}_j + \boldsymbol{p}_j^d], \tag{9}$$

with the neighbor's nodes $\boldsymbol{g}_j$ from $\boldsymbol{v}_i$ found using Dijkstra-like search over the manifold. The weight for each vertex is defined as

$$w_j(\boldsymbol{v}_i) = (1 - ||\boldsymbol{v}_i - \boldsymbol{g}_j||/d_{max}), \tag{10}$$

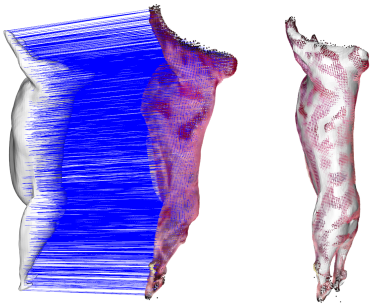
where d_{max} is the maximum distance of the vertex to $\mu+1$ nearest node from $\mathcal{G}$.

Following the example from Fig. 2, the result of posture alignment is shown in Fig. 3. As suggested by Eq. (9), ED



(a) (b)

Fig. 3: In (a), the template and the target are displayed with the skeletons pairwise associations. In (b), the template with corrected posture is superimposed on the target.



(a) (b)

Fig. 4: In (a), the template and the target meshes are displayed with the associations of the surfaces. In (b), the morphed template is superimposed on the target.

works directly on the vertices. As a result, the transformation of $\mathcal{S}_1$ into $\mathcal{S}_1^*$ updates the vertices position such that $\mathcal{S}_1^* = \{\mathcal{V}_1^*, \mathcal{F}_1\}$.

B. Dense surface alignment

The second part of the surface morphism consists of aligning the surface of $\mathcal{S}_1^*$ and $\mathcal{S}_2$. Once the posture is aligned, this task is equivalent to inflating or deflating the surface of $\mathcal{S}_1^*$ to match $\mathcal{S}_2$.

1) Dense surface point association: A local search is performed for the surface vertex association by considering the similarity of the normal orientation with a regularization term on the distance between the vertices.

More formally, the normals $\boldsymbol{o}_{1,j}^* \in \mathcal{O}_1^*$ and $\boldsymbol{o}_{2,j} \in \mathcal{O}_2$ are first computed for the vertices of $\mathcal{S}_1^*$ and $\mathcal{S}_2$ respectively. Then, a set of points is selected by subsampling $\mathcal{V}_1^*$ and for each vertex $\boldsymbol{v}_i^*$ of this subset we find the vertex in $\mathcal{V}_2$ that satisfies the following relation:

$$\boldsymbol{v}_2 = \underset{j}{\operatorname{argmin}}\big(\lambda \left\|\boldsymbol{v}_i^* - \boldsymbol{v}_j\right\|_2 - \boldsymbol{o}_i^*.\boldsymbol{o}_j\big). \tag{11}$$

To reduce the complexity of the search, we transform $\mathcal{V}_2$ into a *Kd-tree* and evaluate Eq. (11) over a subset of the closest

TABLE I: Comparison of our approach with state-of-the-art methods. The maximum error for a variety of noise values is reported (in m). The relative volume and surface area of annotated body part are used as a benchmark.

	average max error $\sigma = 0mm$	***average max error*** $\sigma = 2mm$	***average max error*** $\sigma = 4mm$	***average max error*** $\sigma = 8mm$	***average max error*** $\sigma = 16mm$	***relative volume RMSE***	***relative surface RMSE***	***computation time (s)***
FMaps [13]	1.45	1.54	1.57	1.59	1.70	0.33	0.48	43.12
FMaps + slanted [14]	0.76	0.79	0.82	1.33	1.71	0.52	0.40	**25.08**
FMaps + structured [15]	0.79	0.79	1.06	1.47	1.70	0.36	0.39	36.52
FMaps + BCICP [16]	0.28	0.28	0.30	0.32	0.37	0.31	0.10	562.63
Kernel Matching [17]	1.06	1.05	1.08	1.03	1.24	0.56	0.62	141.00
proposed method	**0.08**	**0.09**	**0.09**	**0.11**	**0.11**	**0.14**	**0.09**	64.96

points. We then stack the dense surface association into two set $\mathcal{A}_1^d$ and $\mathcal{A}_2^d$. An example of the surface pairwise associations is shown in Fig. 4(a).

2) Association filtering: Given the crude definition of Eq. (11), we propose a filter that can trim the associations. Our filter is inspired by the ARAP deformation method [5] and aims to trim a set of associations $(\mathcal{A}_1^d, \mathcal{A}_2^d)$ to minimize the ARAP residuals. Our method is as follows: a) A graph is built on the template from the nodes $\mathcal{A}_1^d$ and connected to their k closest points, this gives us a graph $G_1 = \{\mathcal{A}_1^d, \mathcal{E}_1^d\}$. b) We then transfer the edges $\mathcal{E}_1^d$ on the target to generate the following graph $G_2 = \{\mathcal{A}_2^d, \mathcal{E}_1^d\}$. c) The local rigidity residuals terms are then computed as

$$\mathbf{r}(i) = \sum_{j \in \mathcal{N}(i)} \left\| (\boldsymbol{a}_{2,i} - \boldsymbol{a}_{2,j}) - \hat{\boldsymbol{R}}_i (\boldsymbol{a}_{1,i} - \boldsymbol{a}_{1,j}) \right\|^2 \frac{1}{\|\boldsymbol{a}_{1,i} - \boldsymbol{a}_{1,j}\|^2} \tag{12}$$

where the right part of the equation makes it scale invariant, and d) we then trim the association with the highest residual. This procedure is repeated iteratively until the following termination criterion is met: $\max(\mathbf{r}) < \text{mean}(\mathbf{r}) + 2 * \text{std}(\mathbf{r})$.

3) Surface non-rigid deformation: Given these point associations, the non-rigid deformation is then performed by applying the method described in Section II-A3 by replacing the E_{con} by E_{plane}, to minimize the point to plane distances. The cost function optimized is then defined as follow:

$$E = w_{plane} E_{plane} + w_{rot} E_{rot} + w_{reg} E_{reg} + w_{rig} E_{rig} \tag{13}$$

with E_{plane} defined as

$$E_{plane} = \sum_{l=1}^{n+m} \|\boldsymbol{o}_j . (\boldsymbol{v}_i - \boldsymbol{v}_j)\|_2 . \tag{14}$$

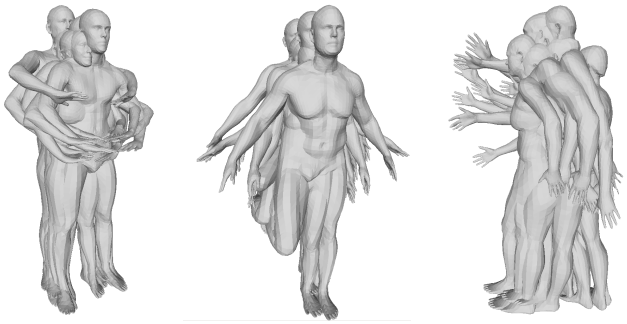
The output of $\mathcal{S}_1^{**} = \{\mathcal{V}_1^{**}, \mathcal{F}_1\}$ is shown in Fig. 4(b).

III. Experiment

A. controlled evaluation

In this section, we present a quantitative evaluation by benchmarking the proposed approach alongside with [13], [14], [15], [16], [17] on the FAUST dataset [18] which provides 3D meshes of ten people posing in a variety of different postures.

The evaluation on the FAUST dataset is performed by considering all the poses which have less than two axes of symmetry (with respect to the skeleton) as shown in Figure 5.



(a) pose 1 (b) pose 2 (c) pose 3

Fig. 5: Samples from the FAUST dataset, used for evaluation. While having similar poses, there are differences in term of shape and scale. A Gaussian noise is added to the surface to emulate a realistic scenario.

The first actor in the dataset is chosen as the template, and all the other actors are set as targets, repeating these experiments for six different postures with a range of different noise added over the surface (in total, our experiment results in 270 different scenarios). In order to simulate the noise generated by RGB-D cameras, we move each vertex by a random distance $d \sim \mathcal{N}(0, \sigma^2)$ in the direction of the vertex normal, with σ chosen from the range $\{0, 2, 4, 8, 16\}$ (in mm). The maximum errors of the shape correspondence between the morphed shape and the ground truth are reported in Table I (we report the mean value of the maximum error over the 270 different scenarios). As an additional evaluation criterion, we also consider the relative RMSE of the volumes after slicing the 3D shapes based on the annotation transfer with $\sigma = 8mm$ for the legs, arms, and torso. The computation time, obtained using a processor Intel i7-4771 - 3.50Ghz, is intended to give an order of magnitude of the computational complexity. However, no particular effort was spent towards code optimisation such as parallelism in our proposed method.

B. Field experiment

The figures used in the methodology show the matching between a template collected using CT and reconstructed using itk-snap [19] and a target reconstructed using a dedicated rig equipped with Primesense RGB-D camera [3].

The proposed approach is further validated using reconstructed meshes from CT data. Four lamb carcasses recon-

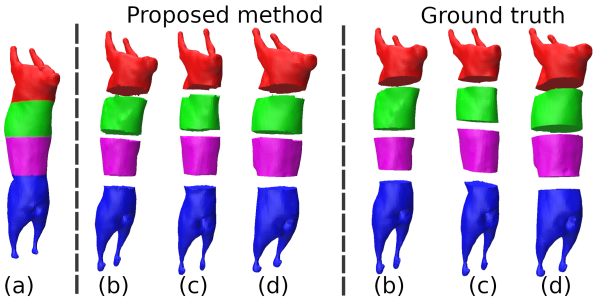


Fig. 6: Example of annotations transfer and virtual cutting. The template (a) is manually annotated. The template is then morphed into the targets (b, c, and d) and the annotation is then used for virtual cutting.

struction where manually annotated by an expert using CT data as a guideline. The annotations shown in Fig. 6 separate the carcasses above the ilio-sacral junction, between the 12^{th} and the 13^{th} ribs, and between the 4^{th} and 5^{th} ribs (these cuts are used in industrial processes [1]). Our method has an average relative accuracy of 93.9% on the volumes of the parts when compared to the ground truth.

IV. Discussion

This paper proposes a method for automatically aligning, morphing, and solving the annotation transfer between two shapes (meshes or point clouds), while handling differences in shape, scale, and rotation. Given an annotated template, this method allows transposing the annotations to a multitude of other targets robustly. The performances of the proposed method are evaluated on two datasets using FAUST [18] for benchmarking our approach with state-of-the-art methods. Further evaluation is performed on carcasses reconstructed from CT and RGB-D data.

Given that our proposed method divide the problem of shape correspondence into a coarse alignment followed by a refinement method, a good generalisation for different breeds of lambs should be expected (i.e., there are no reason why the skeleton extraction, nor the refinement step would fail). However, this has not been thoroughly tested in our experiment, which solely contain crossbred lambs from the MLA Resource Flock [**?**].

The proposed method can solve detecting cutting lines on lamb carcasses. To fully automate cutting of carcass primals, two further works are required. First, a reprojection of the cutting lines in 3D space would be needed. Secondly, the proposed perception system would have to be linked with the control of the robotic arm handling the band-saw. While the former would be trivial, given 3D cameras information, the integration with a robotic arm is more challenging as it requires the fusion of the perception and the control algorithms.

Acknowledgment

This paper is supported by funding from the Australian Government Department of Agriculture and Water Resources as part of its Rural R&D for Profit program, MLA grant number V.RDP.2005.

References

[1] S. Starling, "JBS Fully Automated X-ray Lamb Middle System," Scott Technology, published by Meat and Livestock Australia, Tech. Rep., June 2014.

[2] A. Bab-Hadiashar and A. T. Nama, "Roadmap development for a meat processing intelligent automation centre," RMIT University, published by Australian Meat Processing Corporation, Tech. Rep., 2018.

[3] A. Alempijevic, T. Vidal-Calleja, R. Falque, P. Quin, E. Toohey, B. Walmsley, and M. McPhee, "Lean meat yield estimation using a prototype 3d imaging approach," *Meat Science*, p. 108470, 2021.

[4] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," *ACM Transactions on Graphics*, 2007.

[5] O. Sorkine and M. Alexa, "As-Rigid-As-Possible Surface Modeling," *Eurographics symposium on Geometry processing (SGP)*, 2007.

[6] A. Tagliasacchi, I. Alhashim, M. Olson, and H. Zhang, "Mean curvature skeletons," *Eurographics Symposium on Geometry Processing*, vol. 31, no. 5, pp. 1735–1744, 2012.

[7] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su, "Point Cloud Skeletons via Laplacian Based Contraction," *Shape Modeling International Conference*, pp. 187–197, 2010.

[8] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[9] J. Mitchell and C. H. Mount, David M.Papadimitriou, "The discrete geodesic problem," *SIAM Journal on Computing*, vol. 16, no. 4, pp. 647–668, 1987.

[10] K. Crane, C. Weischedel, and M. Wardetzky, "Geodesics in heat: A new approach to computing distance based on heat flow," *ACM Transactions on Graphics*, vol. 32, no. 3, p. 10, 2013.

[11] C. Jiawen, I. Shahram, and F. Andrew, "KinEtre: Animating the World with the Human Body," *ACM symposium on User interface software and technology (UIST)*, pp. 435–444, 2012.

[12] R. Tarjan, "A note on finding the bridges of a graph," *Information Processing Letters*, vol. 2, no. 6, pp. 160–161, 1974.

[13] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: a flexible representation of maps between shapes," *ACM Transactions on Graphics*, vol. 31, pp. 30:1–30:11, 2012.

[14] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers, "Partial Functional Correspondence," *Computer Graphics Forum*, vol. 36, no. 1, pp. 222–236, 2017.

[15] J. Ren, M. Panine, P. Wonka, and M. Ovsjanikov, "Structured Regularization of Functional Map Computations," *Computer Graphics Forum*, vol. 38, no. 5, pp. 39–53, 2019.

[16] J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov, "Continuous and orientation-preserving correspondences via functional maps," *ACM Transactions on Graphics*, vol. 37, no. 6, 2018.

[17] M. Vestner, Z. Lähner, A. Boyarski, O. Litany, R. Slossberg, T. Remez, E. Rodolà, A. Bronstein, M. Bronstein, R. Kimmel *et al.*, "Efficient deformable shape correspondence via kernel matching," *International Conference on 3D Vision (3DV)*, pp. 517–526, 2017.

[18] F. Bogo, J. Romero, M. Loper, and M. J. Black, "FAUST: Dataset and evaluation for 3D mesh registration," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[19] P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. Gee, and G. Gerig, "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *NeuroImage*, vol. 31, no. 3, pp. 1116–1128, jul 2006.